

# Aula 2 - Isomorfismos e Representações de Grafos

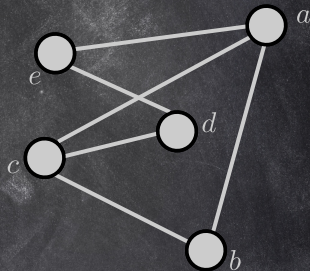
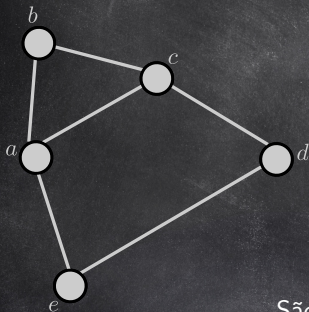
Luís Felipe

UFF

12 de Março de 2026

# O problema do Isomorfismo

Observe os seguintes grafos:

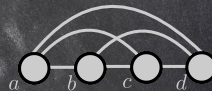
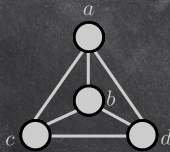
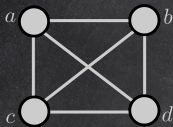


São distintos?  
Não!!

- $G = (V, E)$ , onde  $V = \{a, b, c, d, e\}$  e  $E = \{ab, ac, ae, bc, cd, de\}$

# O problema do Isomorfismo

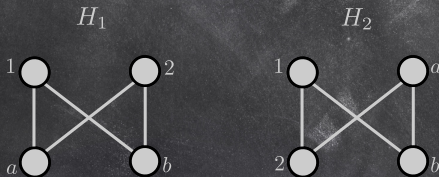
**Exemplo 1:** Observe algumas representações para o  $K_4$ .



## O problema do Isomorfismo

Apesar de duas **representações geométricas** distintas poderem associar a um mesmo grafo, duas representações geométricas iguais podem representar grafos distintos.

Exemplo 2:



$V(H_1) = V(H_2) = \{1, 2, a, b\}$ , mas  
 $E(H_1) = \{1a, 1b, 2a, 2b\} \neq \{12, 1b, 2a, ab\} = E(H_2)$

Porém, se **renomeássemos os vértices** (trocando 2 com  $a$ ), voltaríamos a ter o mesmo grafo.

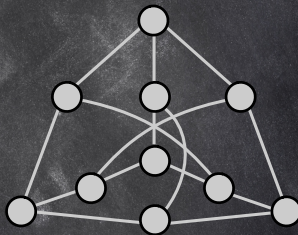
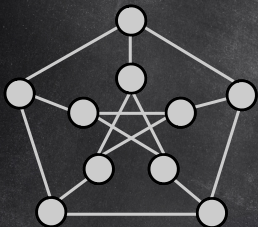
# O problema do Isomorfismo

- Dadas duas representações geométricas, elas correspondem a um mesmo grafo?
- Ou seja, é possível fazer as duas representações coincidirem de modo a preservar as adjacências?

Dois grafos  $G_1 = (V_1, E_1)$  e  $G_2 = (V_2, E_2)$  são **isomorfos** e denotamos por  $G_1 \approx G_2$  se existir uma **função bijetora**  $f : V_1 \rightarrow V_2$  que satisfaça:

$$ab \in E_1 \text{ se e somente se } f(a)f(b) \in E_2.$$

## Exemplo



- Esses dois grafos são isomorfos? **Sim!!!** Por que?

# Problema do Isomorfismo

Dados grafos  $G_1$  e  $G_2$ ,  $G_1 \approx G_2$ ? Ainda não se sabe a complexidade deste problema, sendo ele, portanto, um problema em aberto.

- O problema de isomorfismo em grafos é uma restrição do problema de isomorfismo de subgrafos. Este sim é sabidamente difícil de resolver computacionalmente. (Obs.: Conteúdo de ASA!!!)

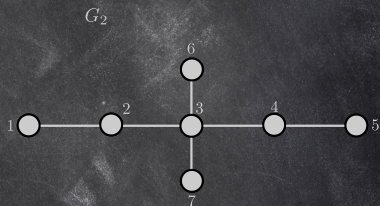
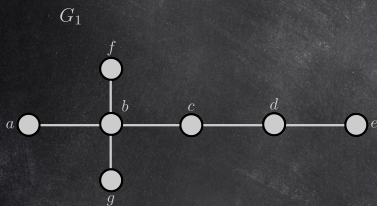
Contudo, podemos listar algumas condições que são necessárias para que dois grafos sejam considerados isomorfos:

- Mesma quantidade de vértices;
- Mesma quantidade de arestas;
- Mesma quantidade de vértices com um mesmo grau.

Essas condições são suficientes???

# Problema do Isomorfismo

**Exemplo:** Os grafos  $G_1$  e  $G_2$

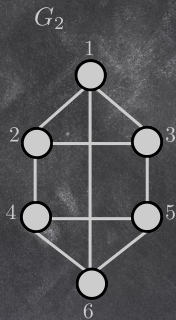
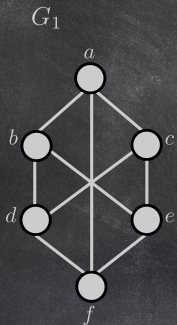


são isomorfos?

**Não**, pois se existir uma função de isomorfismo, ela mapeia o vértice  $b$  no vértice 3 e não existem, em  $G_2$ , 3 vértices de grau 1 adjacentes a 3.

# Problema do Isomorfismo

**Exemplo:** Os grafos  $G_1$  e  $G_2$



são isomorfos?

**Exercício!** Dica: repare nas arestas 12, 23, 13.

# Representações de um grafo

- Na Aula 1 vimos que podemos representar um grafo:
  - ▶ Através da enumeração de seus vértices e arestas, e
  - ▶ Através da representação geométrica no plano.
- Mas como podemos representar um grafo no computador? Veremos 3 maneiras:
  - ▶ Matriz de adjacência,
  - ▶ Matriz de incidência, e
  - ▶ Lista de adjacência

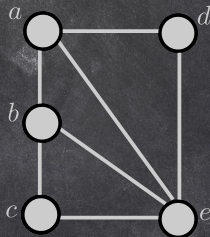
## Matriz de Adjacência

Dado um grafo  $G = (V, E)$ , onde  $V = \{v_1, \dots, v_n\}$ , a **matriz de adjacência**  $G$  é uma matriz  $n \times n$  tal que:

$$A_{n \times n} = [a_{ij}] = \begin{cases} a_{ij} = 1, & \text{se } v_i v_j \in E; \\ a_{ij} = 0, & \text{caso contrário.} \end{cases}$$

Exemplo 1:

|          | <i>a</i> | <i>b</i> | <i>c</i> | <i>d</i> | <i>e</i> |
|----------|----------|----------|----------|----------|----------|
| <i>a</i> | 0        | 1        | 0        | 1        | 1        |
| <i>b</i> | 1        | 0        | 1        | 0        | 1        |
| <i>c</i> | 0        | 1        | 0        | 0        | 1        |
| <i>d</i> | 1        | 0        | 0        | 0        | 1        |
| <i>e</i> | 1        | 1        | 1        | 1        | 0        |



# Propriedades

A **matriz de adjacência** de um grafo

- Possui diagonal principal nula.
- É simétrica com relação à diagonal principal.
- A soma de cada linha ou coluna fornece o grau do vértice associado.
- Possui  $2m$  entradas não nulas
- uma matriz de adjacência caracteriza **univocamente** um grafo.

# Propriedades

**OBS.:** Diversas matrizes de adjacência podem corresponder a um mesmo grafo  $G$ . Basta permutar os vértices.

**Exemplo 2:**  $(a, b, c, d, e) \rightarrow (c, d, a, b, e)$

|     | $a$ | $b$ | $c$ | $d$ | $e$ |
|-----|-----|-----|-----|-----|-----|
| $a$ | 0   | 1   | 0   | 1   | 1   |
| $b$ | 1   | 0   | 1   | 0   | 1   |
| $c$ | 0   | 1   | 0   | 0   | 1   |
| $d$ | 1   | 0   | 0   | 0   | 1   |
| $e$ | 1   | 1   | 1   | 1   | 0   |

$\rightarrow$

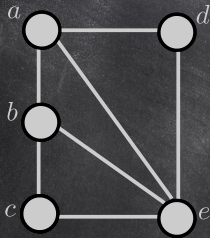
|     | $c$ | $d$ | $a$ | $b$ | $e$ |
|-----|-----|-----|-----|-----|-----|
| $c$ | 0   | 0   | 0   | 1   | 1   |
| $d$ | 0   | 0   | 1   | 0   | 1   |
| $a$ | 0   | 1   | 0   | 1   | 1   |
| $b$ | 1   | 0   | 1   | 0   | 1   |
| $e$ | 1   | 1   | 1   | 1   | 0   |

## Matriz de Incidência

Dado um grafo  $G = (V, E)$ , onde  $V = \{v_1, \dots, v_n\}$  e  $E = \{e_1, \dots, e_m\}$ , a **matriz de incidência**  $G$  é uma matriz  $n \times m$  tal que:

$$M_{n \times m} = [m_{ij}] = \begin{cases} a_{ij} &= 1, \text{ se } e_j \text{ é incidente a } v_i; \\ a_{ij} &= 0, \text{ caso contrário.} \end{cases}$$

### Exemplo 3:



|     | $ab$ | $ad$ | $ae$ | $bc$ | $be$ | $ce$ | $de$ |
|-----|------|------|------|------|------|------|------|
| $a$ | 1    | 1    | 1    | 0    | 0    | 0    | 0    |
| $b$ | 1    | 0    | 0    | 1    | 1    | 0    | 0    |
| $c$ | 0    | 0    | 0    | 1    | 0    | 1    | 0    |
| $d$ | 0    | 1    | 0    | 0    | 0    | 0    | 1    |
| $e$ | 0    | 0    | 1    | 0    | 1    | 1    | 1    |

# Propriedades

A **matriz de incidência** de um grafo

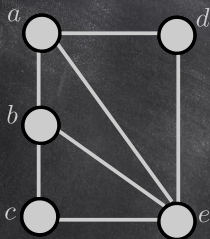
- Possui  $2m$  entradas não nulas
- Uma matriz de incidência caracteriza univocamente um grafo.

## Matriz aresta $\times$ aresta

Dado um grafo  $G = (V, E)$ , onde  $E = \{e_1, \dots, e_m\}$ , a **matriz aresta  $\times$  aresta** de  $G$  é uma matriz  $m \times m$  tal que:

$$C_{m \times m} = [c_{ij}] = \begin{cases} c_{ij} &= 1, \text{ se } e_i \text{ é adjacente a } e_j; \\ c_{ij} &= 0, \text{ caso contrário.} \end{cases}$$

### Exemplo 4:



|      | $ab$ | $ad$ | $ae$ | $bc$ | $be$ | $ce$ | $de$ |
|------|------|------|------|------|------|------|------|
| $ab$ | 0    | 1    | 1    | 1    | 1    | 0    | 0    |
| $ad$ | 1    | 0    | 1    | 0    | 0    | 0    | 1    |
| $ae$ | 1    | 1    | 0    | 0    | 1    | 1    | 1    |
| $bc$ | 1    | 0    | 0    | 0    | 1    | 1    | 0    |
| $be$ | 1    | 0    | 1    | 1    | 0    | 1    | 1    |
| $ce$ | 0    | 0    | 1    | 1    | 1    | 0    | 1    |
| $de$ | 0    | 1    | 1    | 0    | 1    | 1    | 0    |

Quantas entradas não nulas??  $\sum_{v \in V(G)} d(v)(d(v) - 1)$ .

Por que?

Tome um vértice  $v \in V(G)$ . Cada aresta com extremo em  $v$  possui  $d(v) - 1$  vizinhos com extremos a  $v$ . Fazemos essa contagem para todos os vértices de  $G$ .

**Obs.:** Veja questão 11 da lista 1.

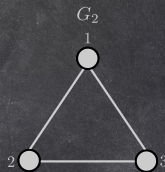
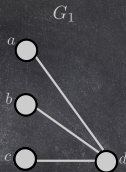
# Propriedades

A matriz **aresta**  $\times$  **aresta** de um grafo

- Possui diagonal principal nula,
- É simétrica com relação à diagonal principal,
- **NÃO** representa um grafo!!!! Existem grafos **não** isomorfos com a mesma matriz  $C$ .

Exemplo 5:

$$C = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$



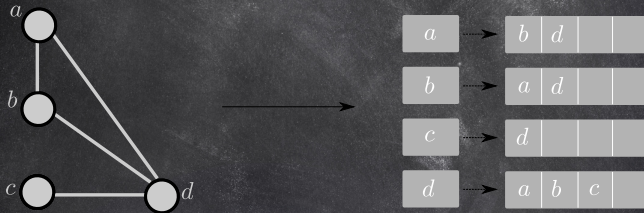
$G_1 \not\cong G_2$

# Lista de Adjacência

- Em uma **Lista de Adjacência** cada vértice está associado a um vetor de dimensão  $n$  (número de vértices no grafo)
- Cada vértice possui uma lista de vértices adjacentes

# Lista de Adjacência

## Exemplo 6:



# Matriz $\times$ Lista de Adjacência

- Qual a melhor estrutura de dados para representar um grafo em um computador?

Para responder a esta pergunta, vamos descrever as desvantagens de cada tipo de representação.

- ▶ **Desvantagem da representação matricial:**

Com grafos esparsos, a matriz é formada principalmente de zeros, o que gera grande consumo de memória **desnecessário**.

- ▶ **Desvantagem da representação por listas:**

Em grafos onde os vértices têm muitos vizinhos (mas ainda bem menos do que  $n$ ), as listas ficam longas, o que pode aumentar o tempo de acesso.

## Vantagens / Desvantagens

| Tempo de execução            | Matriz | Lista       |
|------------------------------|--------|-------------|
| Inserir aresta               | $O(1)$ | $O(1)$      |
| Remover aresta               | $O(1)$ | $O(\Delta)$ |
| Testar adjacência            | $O(1)$ | $O(\Delta)$ |
| Listar vizinhos de $v \in V$ | $O(n)$ | $O(\Delta)$ |

Logo, a melhor estrutura depende do algoritmo!

# Implementação

## Matriz de adjacência:

```
1 n, m = map(int, input().split())
2
3 mat = [[0] * n for _ in range(n)]
4
5 for _ in range(m):
6     u, v = map(int, input().split())
7     u -= 1
8     v -= 1
9     mat[u][v] = 1
10    mat[v][u] = 1
11
12 for linha in mat:
13     for valor in linha:
14         print(valor, end=" ")
15     print()
```